BioSystems 94 (2008) 95–101

Contents lists available at ScienceDirect

BioSystems

journal homepage: www.elsevier.com/locate/biosystems

A cell pattern generation model based on an extended artificial regulatory network

Arturo Chavoya [a,c,*], Yves Duthen [b,c]

[a] Universidad de Guadalajara, Periférico Norte 799, Zapopan, Jal., CP 45000, Mexico
[b] Université de Toulouse 1, 1 Place Anatole France, 31000 Toulouse, France
[c] Institut de Recherche en Informatique de Toulouse (IRIT), VORTEX Team, 118 Route de Narbonne, 31062 Toulouse, France

ARTICLE INFO

Article history:
Received 7 May 2007
Received in revised form 30 October 2007
Accepted 23 May 2008

Keywords:
Cell pattern
Artificial regulatory network
Genetic algorithms
French flag problem
Cellular automata

ABSTRACT

Cell pattern generation has a fundamental role in both artificial and natural development. This paper presents results from a model in which a genetic algorithm (GA) was used to evolve an artificial regulatory network (ARN) to produce predefined 2D cell patterns through the selective activation and inhibition of genes. The ARN used in this work is an extension of a model previously used to create simple geometrical patterns. The GA worked by evolving the gene regulatory network that was used to control cell reproduction, which took place in a testbed based on cellular automata (CA). After the final chromosomes were produced, a single cell in the middle of the CA lattice was allowed to replicate controlled by the ARN found by the GA, until the desired cell pattern was formed. The model was applied to the problem of generating a French flag pattern.

1. Introduction

Computational Development is a relatively new sub-field of Evolutionary Computation that studies artificial models of cellular growth, with the objective of understanding how complex structures and patterns can emerge from a small group of initial undifferentiated cells (Kumar and Bentley, 2003). In biological systems, development is a fascinating and very complex process that involves following an extremely intricate program coded in the organism's genome.

One of the crucial stages in the development of an organism is that of pattern formation, where the fundamental body plans of the individual are delineated. It is now evident that gene regulatory networks play a central role in the development and metabolism of living organisms (Davidson, 2006). Furthermore, it has been found in recent years that the different cell patterns created during the development of an organism are mainly due to the selective activation and inhibition of very specific regulatory genes.

Artificial regulatory networks (ARNs) are computer models whose objective is to mimic to some extent the gene regulatory networks found in nature. ARNs have previously been used to study differential gene expression either as a computational paradigm or to solve particular problems (Eggenberger, 1997; Reil, 1999; Banzhaf, 2003; Kuo and Banzhaf, 2004; Stewart et al., 2005; Flann et al., 2005). On the other hand, evolutionary computation techniques have been extensively used in the past in a wide range of applications, and in particular they have previously been used to evolve ARNs to perform specific tasks (Bongard, 2002; Kuo et al., 2004).

In this paper we describe results on the use of a genetic algorithm (GA) to evolve an ARN in order to create predefined 2D patterns by means of the selective activation and inhibition of genes. The ARN used in this work is an extension of the model developed by Banzhaf (2003). We decided to extend the model because in previous work we ran into limits in the number of regulatory genes that could be reliably synchronized under the conditions essayed (Chavoya and Duthen, 2007). In order to test the functionality of the ARN found by the GA, we applied the chromosomes representing the ARN to a cellular growth model that we have successfully used in the past to develop simple 2D and 3D geometrical shapes (Chavoya and Duthen, 2006b).

The paper starts with a section describing the French flag problem with a brief description of models that have used it as a test case. The next section describes the cellular growth testbed developed to evaluate the evolved ARNs, followed by a section presenting the ARN model and how it was implemented. The following section describes the GA used and how it was applied to evolve the ARN. Results are presented next, followed by a section of conclusions.

* Corresponding author at: Universidad de Guadalajara, Periférico Norte 799, Zapopan, Jal., Mexico CP 45000, Mexico.
E-mail addresses: achavoya@cucea.udg.mx (A. Chavoya), yves.duthen@univ-tlse1.fr (Y. Duthen).

0303-2647/$ – see front matter © 2008 Elsevier Ireland Ltd. All rights reserved.
doi:10.1016/j.biosystems.2008.05.015

2. The French Flag Problem

The problem of generating a French flag pattern was first introduced by Wolpert in the late 1960s when trying to formulate the problem of cell pattern development and regulation in living organisms (Wolpert, 1968). This formulation has been used since then by some authors to study the problem of artificial pattern development.

Lindenmayer and Rozenberg (1972) used the French flag problem to illustrate how a grammar-based L-System could be used to solve the generation of this particular pattern when enunciated as the production of a string of the type $a^n b^n c^n$ over the alphabet $\{a, b, c\}$ and with $n > 0$. On the other hand, Herman and Liu (1973) developed an extension of a simulator called CELIA (Baker and Herman, 1970) and applied it to generate a French flag pattern in order to study synchronization and symmetry breaking in cellular development.

More recently, Miller and Banzhaf (2003) used what they called Cartesian genetic programming to evolve a cell program that would construct a French flag pattern. They tested the robustness of their programs by manually removing parts of the developing pattern. They found that some of their evolved programs could repair to some extent the damaged patterns. Bowers (2005) also used this problem to study the phenotypic robustness of his embryogeny model, which was based on cellular growth with diffusing chemicals as signaling molecules.

Gordon and Bentley (2005) proposed a development model based on a set of rules that described how development should proceed. A set of rules evolved by a GA was used to develop a French flag pattern. The morphogenic model based on a multiagent system developed by Beurier et al. (2006) also used an evolved set of agent rules to grow French and Japanese flag patterns. On the other hand, Devert et al. (2007) proposed a neural network model for multicellular development that grew French flag patterns.

Finally, even models for developing evolvable hardware have benefited from the French flag problem as a test case (Tyrrell and Greensted, 2007; Harding et al., 2007).

3. Cellular Growth Testbed

Cellular automata (CA) were chosen as models of cellular growth, since they provide a simple mathematical model that can be used to study self-organizing features of complex systems (Wolfram, 1983). CA are characterized by a regular lattice of N identical cells, an interaction neighborhood template η, a finite set of cell states Σ, and a space- and time-independent transition rule ϕ which is applied to every cell in the lattice at each time step.

In the cellular growth testbed used in this work, a 33×33 regular lattice with non-periodic boundaries was used. The set of cell states was defined as $\Sigma = \{0, 1\}$, where 0 can be interpreted as an empty cell and 1 as an occupied or active cell. The interaction template η used was an outer Moore neighborhood. The CA rule ϕ was defined as a lookup table that determined, for each local neighborhood, the state (empty or occupied) of the objective cell at the next time step. For a binary-state CA, these update states are termed the rule table's "output bits". The lookup table input was defined by the binary state value of cells in the local interaction neighborhood, where 0 meant an empty cell and 1 meant an occupied cell (Chavoya and Duthen, 2006a). A cell can become active only if there is already an active cell in the interaction neighborhood. Thus, a new active cell can only be derived (reproduced) from a previously activated cell in the interaction neighborhood. Starting with an active cell in the middle of the lattice, the CA algorithm is applied allowing active cells to reproduce for 100 time steps according to the CA rule table. During an iteration of the CA algorithm, the order of reproduction of active cells is randomly selected in order to avoid artifacts caused by a deterministic order of cell reproduction. For the sake of simplicity, cell death is not considered in the present model.

For all experiments, the CA were implemented as NetLogo models. NetLogo is a programmable modeling environment based on StarLogo that can be used to simulate natural and social phenomena (Wilensky, 1999). It works by giving instructions to hundreds or thousands of independent "agents" all operating concurrently. It is well suited to study emergent properties in complex systems that result from the interaction of simple but often numerous entities. For each of the cell patterns studied, a NetLogo model was built.

4. Artificial Regulatory Network

An artificial regulatory network is a gene control model inspired by its biological counterpart. In nature, gene regulatory networks are widely used to control development and metabolic functions in living organisms (Davidson, 2006). Biological gene regulatory networks are a central component of an organism's genome, which is coded as one or more chains of DNA, and that interact with other macromolecules, such as RNA and proteins. On the other hand, artificial genomes are usually coded as strings of discrete data types. The genome used in this work was implemented as a binary string starting with a series of regulatory genes that make up an ARN, followed by a number of structural genes (see Fig. 1).

The gene regulatory network implemented in this work is an extension of the ARN presented in Chavoya and Duthen (2007), which in turn is based on the model proposed by Banzhaf (2003). However, unlike the ARN developed by this author, the genome implemented in the present work does not have promoter sequences and there are no unused intergene regions. All regulatory genes are adjacent and have predefined initial and end positions. Furthermore, the number of regulatory genes is fixed.

The original model only considered one inhibitor and one enhancer site for each regulatory gene (Banzhaf, 2003). However, in the present model the number of regulatory sites can be more than two and, more significantly, they have no predefined function. They can behave either as an enhancer or an inhibitor, depending on the configuration of the function defining bits associated with the regulatory site (Fig. 1). If there are more 1's than 0's in the function defining region, then the site functions as an enhancer, but if there are more 0's than 1's, then the site behaves as an inhibitor. Finally, if there is an equal number of 1's and 0's, then the regulatory site is turned off. This means that the regulatory site role as an enhancer or as an inhibitor can be evolved by the GA. Furthermore, if the number of function defining bits is even, then the regulatory site can be turned on and off. The number of regulatory sites was extended with respect to the original model in order to more closely follow what happens in nature, where biological regulatory genes involved in development typically have several regulatory sites associated with them (Davidson, 2006).

As mentioned above, the ARN used in the present work consists of a series of regulatory genes, each of which consists in turn of a series of inhibitor/enhancer sites and a series of regulatory protein coding regions (Fig. 1). The latter "translate" a protein using the majority rule, i.e. for each bit position in the protein coding regions, the number of 1's and 0's is counted and the bit that is in majority is translated into the regulatory protein.

The regulatory sites and the individual protein coding regions all have the same size in bits. Thus the protein translated from the coding regions can be compared on a bit by bit basis with the reg-